

[illegible]

EXE

Mod

ED 1

ED

ED

ED

ED

ED1

ED

ED

ED
EDED
EDSYS
LIB

L18

```
CCCCCCCC HH HH MM MM PPPPPPP AAAAAA SSSSSSSS TTTTTTTTTT EEEEEEEEEEE
CCCCCCCC HH HH MM MM PPPPPPP AAAAAA SSSSSSSS TTTTTTTTTT EEEEEEEEEEE
CC CC HH HH MMMM MMMM PP PP AA AA SS SSSSSS TT TT EEEEEEEEEEE
CC CC HH HH MMMM MMMM PP PP AA AA SS SSSSSS TT TT EEEEEEEEEEE
CC CC HH HH MM MM MM PP PP AA AA SS SSSSSS TT TT EEEEEEEEEEE
CC CC HHHHHHHHHH MM MM PPPPPPP AAAAAA SSSSSS SS TT TT EEEEEEEEEEE
CC CC HHHHHHHHHH MM MM PPPPPPP AAAAAA SSSSSS SS TT TT EEEEEEEEEEE
CC CC HH HH MM MM PP PP AAAAAAAAAA SS TT TT EEEEEEEEEEE
CC CC HH HH MM MM PP PP AAAAAAAAAA SS TT TT EEEEEEEEEEE
CC CC HH HH MM MM PP PP AAAAAAAAAA SS TT TT EEEEEEEEEEE
CC CC HH HH MM MM PP PP AAAAAAAAAA SS TT TT EEEEEEEEEEE
CCCCCCCC HH HH MM MM PP AA AA SSSSSSSS TT EEEEEEEEEEE
CCCCCCCC HH HH MM MM PP AA AA SSSSSSSS TT EEEEEEEEEEE
```

```
LL LL I I I I I SSSSSSSS
LL LL I I I I I SSSSSSSS
LL LL I I SS
LL LL I I SS
LL LL I I SS
LL LL I I SSSSSS
LL LL I I SSSSSS
LL LL I I SS
LL LL I I SS
LL LL I I SS
LLLLLLLLLL I I I I I SSSSSSSS
LLLLLLLLLL I I I I I SSSSSSSS
```

.....


```
0001 0 %TITLE 'EDT$CHMPASTE - change mode PASTE command'
0002 0 MODULE EDT$CHMPASTE (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This module executes the change mode PASTE command.
0037 1
0038 1 ENVIRONMENT: Runs at any access mode - AST reentrant
0039 1
0040 1 AUTHOR: Bob Kushlis, CREATION DATE: Unknown
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. DJS 04-Feb-1981. This module was created by
0045 1 extracting the routine EDT$$PST_CMD from the module CHANGE.BLI.
0046 1 1-002 - Regularize headers. JBS 03-Mar-1981
0047 1 1-003 - Change SPLIT_LINE to EDT$$SPLT_LNINS. JBS 30-Mar-1981
0048 1 1-004 - Use new message codes. JBS 06-Aug-1981 (this note added 24-May-1982)
0049 1 1-005 - Set a flag if control C actually aborts something. JBS 24-May-1982
0050 1 1-006 - New screen update logic. JBS 13-Sep-1982
0051 1 1-007 - Remove EDT$$G_LN_NO for new screen update logic. JBS 29-Sep-1982
0052 1 1-008 - Convert to new line number compare macro. STS 20-Oct-1982
0053 1 1-009 - Add a parameter to the split line routines. SMB 16-Nov-1982
0054 1 1-010 - Remove parameter from EDT$$SPLT_LN. JBS 17-Nov-1982
0055 1 1-011 - Add a parameter to EDT$$COMB_LN. JBS 28-Dec-1982
0056 1 1-012 - Improve efficiency by accumulating blank lines until they are needed. JBS 05-Apr-1983
0057 1 1-013 - Do a single insertion sequence, to avoid problems with sequence numbers. JBS 01-Jun-1983
```

EDT\$CHMPASTE
V04-000

EDT\$CHMPASTE - change mode PASTE command

J 7
16-Sep-1984 00:08:07
14-Sep-1984 12:22:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMPASTE.BLI;1 Page (1) 2

: 58
: 59
: 60
0058 1 ! 1-014 - Decrement the insert count if we end by combining lines. JBS 01-Jun-1983
0059 1 !--
0060 1

EDTSCHMPASTE
V04-000

EDTSCHMPASTE - change mode PASTE command
Declarations

K 7
16-Sep-1984 00:08:07
14-Sep-1984 12:22:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMPASTE.BLI;1 Page 3 (2)

```

: 62      0061 1 %SBTTL 'Declarations'
: 63      0062 1
: 64      0063 1 : TABLE OF CONTENTS:
: 65      0064 1 :
: 66      0065 1
: 67      0066 1 REQUIRE 'EDTSRC:TRAROUNAM';
: 68      0505 1
: 69      0506 1 FORWARD ROUTINE
: 70      0507 1     EDT$SPST_CMD;
: 71      0508 1
: 72      0509 1 :
: 73      0510 1 : INCLUDE FILES:
: 74      0511 1 :
: 75      0512 1
: 76      0513 1 REQUIRE 'EDTSRC:EDTREQ';
: 77      0648 1
: 78      0649 1 :
: 79      0650 1 : MACROS:
: 80      0651 1
: 81      0652 1 :     NONE
: 82      0653 1
: 83      0654 1 : EQUATED SYMBOLS:
: 84      0655 1
: 85      0656 1 :     NONE
: 86      0657 1
: 87      0658 1 : OWN STORAGE:
: 88      0659 1
: 89      0660 1 :     NONE
: 90      0661 1
: 91      0662 1 : EXTERNAL REFERENCES:
: 92      0663 1
: 93      0664 1 :     In the routine
```

! Execute the paste command

EDT
V04

SR
C


```

: 95      0665 1 %SBTTL 'EDT$$PST_CMD - change mode PASTE command'
: 96      0666 1
: 97      0667 1 GLOBAL ROUTINE EDT$$PST_CMD                ! Change mode PASTE command
: 98      0668 1 =
: 99      0669 1
: 100     0670 1 ++
: 101     0671 1 FUNCTIONAL DESCRIPTION:
: 102     0672 1
: 103     0673 1     This routine executes the paste command. The variable EDT$$A_ALT_BUF has
: 104     0674 1     the address of the TBCB for the buffer to be pasted.
: 105     0675 1
: 106     0676 1 FORMAL PARAMETERS:
: 107     0677 1
: 108     0678 1     NONE
: 109     0679 1
: 110     0680 1 IMPLICIT INPUTS:
: 111     0681 1
: 112     0682 1     EDT$$A_ALT_BUF
: 113     0683 1     EDT$$A_CUR_BUF
: 114     0684 1     EDT$$L_LN00
: 115     0685 1     EDT$$L_LN01
: 116     0686 1     EDT$$A_WK_LN
: 117     0687 1
: 118     0688 1 IMPLICIT OUTPUTS:
: 119     0689 1
: 120     0690 1     EDT$$G_PST_CNT
: 121     0691 1     EDT$$G_CC_DONE
: 122     0692 1     EDT$$L_WK_INSCNT
: 123     0693 1
: 124     0694 1 ROUTINE VALUE
: 125     0695 1
: 126     0696 1     A value of 0 is returned if the paste failed.
: 127     0697 1
: 128     0698 1 SIDE EFFECTS:
: 129     0699 1
: 130     0700 1     NONE
: 131     0701 1
: 132     0702 1 --
: 133     0703 1
: 134     0704 2 BEGIN
: 135     0705 2
: 136     0706 2 EXTERNAL ROUTINE
: 137     0707 2     EDT$$INS_STR,
: 138     0708 2     EDT$$MSG_BELL : NOVALUE,
: 139     0709 2     EDT$$CHK_CC,
: 140     0710 2     EDT$$COMB_LN : NOVALUE,
: 141     0711 2     EDT$$CS_DWN,
: 142     0712 2     EDT$$CS_LEFT,
: 143     0713 2     EDT$$CS_RIGHT,
: 144     0714 2     EDT$$SPCT_LN : NOVALUE,
: 145     0715 2     EDT$$END_INS,
: 146     0716 2     EDT$$RD_CURLN,
: 147     0717 2     EDT$$RD_NXTLN,
: 148     0718 2     EDT$$START_INS,
: 149     0719 2     EDT$$TOP_BUF;
: 150     0720 2
: 151     0721 2 EXTERNAL
```

! Insert a string of characters at the current position
! Output a message to the terminal with a warning bell
! Check to see if a CTRL/C has been typed
! Combine the current line with the one immediately above it
! Move down a line
! Move left a character
! Move right a character
! Split a line of text at the current cursor position
! End an insert sequence
! Get the current line
! Move forward a line
! Start an insert sequence
! Go to top of buffer

```
152 0722 2 EDT$$G_PST_CNT,      ! No. of characters pasted.
153 0723 2 EDT$$A_ALT_BUF : REF TBCB_BLOCK, ! Alternate buffer used for cut/paste.
154 0724 2 EDT$$A_CUR_BUF : REF TBCB_BLOCK, ! The current buffer tbc
155 0725 2 EDT$$L_LN00 : LN_BLOCK,          ! Line number 10**0 (1)
156 0726 2 EDT$$A_WK_LN : REF LIN_BLOCK,    ! Current line pointer
157 0727 2 EDT$$G_CC_DONE,                  ! Set to 1 if control C actually aborts something
158 0728 2 EDT$$L_WK_INSCNT : LN_BLOCK;      ! Number of records inserted in this insert sequence
159 0729 2
160 0730 2 MESSAGES ((ATTPASCUR));
161 0731 2
162 0732 2 LOCAL
163 0733 2     SAVE TBCB,
164 0734 2     BLANK_LINES;
165 0735 2
166 0736 2 IF (.EDT$$A_ALT_BUF EQL 0) THEN RETURN (0);
167 0737 2
168 0738 2 !+
169 0739 2 !- Check for paste of the current buffer.
170 0740 2 !-
171 0741 2
172 0742 2 IF (.EDT$$A_ALT_BUF EQL .EDT$$A_CUR_BUF)
173 0743 2 THEN
174 0744 2     BEGIN
175 0745 2     EDT$$MSG_BELL (EDT$_ATTPASCUR);
176 0746 2     RETURN (0);
177 0747 2     END;
178 0748 2
179 0749 2 !+
180 0750 2 !- Save the address of the current text buffer.
181 0751 2 !-
182 0752 2     SAVE TBCB = .EDT$$A_CUR_BUF;
183 0753 2     EDT$$G_PST_CNT = 0;
184 0754 2 !+
185 0755 2 !- Check for empty buffer.
186 0756 2 !-
187 0757 2
188 0758 2 IF (CMLNO (EDT$$L_LN00, EDT$$A_ALT_BUF [TBCB_LINE_COUNT]) EQL 1) THEN RETURN (1);
189 0759 2
190 0760 2 EDT$$START_INS ();          ! Perform initialization for text insertion
191 0761 2 !+
192 0762 2 !- Split the current line, so we can find our current point again.
193 0763 2 !-
194 0764 2     EDT$$SPLT_LN ();
195 0765 2     EDT$$CS_LEFT ();
196 0766 2 !
197 0767 2     BLANK_LINES = 0;
198 0768 2 !+
199 0769 2 !- Position to the front of the paste buffer.
200 0770 2 !-
201 0771 2     EDT$$A_CUR_BUF = .EDT$$A_ALT_BUF;
202 0772 2     EDT$$TOP_BUF ();
203 0773 2
204 0774 2 WHILE 1 DO                  ! Loop through all of the paste buffer.
205 0775 2     BEGIN
206 0776 2 !+
207 0777 2 !- Handle the next line. If the line is of zero length just count the blank lines counter.
208 0778 2 !- Otherwise insert all of the blank lines counted plus this line.
```



```
209 0779 3 !-
210 0780 3
211 0781 4 IF (.EDT$$A_WK_LN [LIN_LENGTH] NEQ 0)
212 0782 3 THEN
213 0783 4 BEGIN
214 0784 4 EDT$$A_CUR_BUF = .SAVE_TBCB;
215 0785 4 EDT$$RD_CURLN ();
216 0786 4
217 0787 4 WHILE (.BLANK_LINES GTR 0) DO
218 0788 5 BEGIN
219 0789 5 EDT$$SPLT_LN ();
220 0790 5 BLANK_LINES = .BLANK_LINES - 1;
221 0791 4 END;
222 0792 4
223 0793 4 EDT$$A_CUR_BUF = .EDT$$A_ALT_BUF;
224 0794 4 EDT$$RD_CURLN ();
225 0795 4 EDT$$INS_STR (EDT$$A_WK_LN [LIN_TEXT], .EDT$$A_WK_LN [LIN_LENGTH]);
226 0796 3 END;
227 0797 3
228 0798 3 EDT$$G_PST_CNT = .EDT$$G_PST_CNT + .EDT$$A_WK_LN [LIN_LENGTH] + 1;
229 0799 3
230 0800 3 IF ( NOT EDT$$RD_NXTLN ()) THEN EXITLOOP;
231 0801 3
232 0802 3 !+
233 0803 3 ! Check for CTRL/C.
234 0804 3 !-
235 0805 3
236 0806 3 IF EDT$$CHK_CC ()
237 0807 3 THEN
238 0808 4 BEGIN
239 0809 4 EDT$$G_CC_DONE = 1;
240 0810 4 EXITLOOP;
241 0811 3 END;
242 0812 3
243 0813 3 BLANK_LINES = .BLANK_LINES + 1;
244 0814 2 END;
245 0815 2
246 0816 2 !+
247 0817 2 ! We have completed the insertion except for possible trailing blank lines. We must also
248 0818 2 ! undo the initial split. If the last line to be inserted was blank we don't need to
249 0819 2 ! undo the initial split.
250 0820 2 !-
251 0821 2 EDT$$A_CUR_BUF = .SAVE_TBCB;
252 0822 2 EDT$$RD_CURLN ();
253 0823 2
254 0824 2 WHILE (.BLANK_LINES GTR 1) DO
255 0825 3 BEGIN
256 0826 3 EDT$$SPLT_LN ();
257 0827 3 BLANK_LINES = .BLANK_LINES - 1;
258 0828 2 END;
259 0829 2
260 0830 2 IF (.BLANK_LINES EQL 0)
261 0831 2 THEN
262 0832 3 BEGIN
263 0833 3 EDT$$CS_DWN ();
264 0834 3 EDT$$COMB_LN ();
265 0835 3 SUBLINE (NUMBER_ONE, EDT$$L_WK_INSCNT);
```


EDT\$CHMPASTE
V04-000

EDT\$CHMPASTE - change mode PASTE command
EDT\$\$PST_CMD - change mode PASTE command

B 8
16-Sep-1984 00:08:07
14-Sep-1984 12:22:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMPASTE.BLI;1
Page 7
(3)

```
: 266      0836 3      EDT$$G_PST_CNT = .EDT$$G_PST_CNT - 1;
: 267      0837 3      END
: 268      0838 2      ELSE
: 269      0839 2      EDT$$CS_RIGHT ();
: 270      0840 2
: 271      0841 2      EDT$$END_INS ();
: 272      0842 3      RETURN (T)
: 273      0843 1      END;
```

! End the text insertion sequence

! of routine EDT\$\$PST_CMD

.TITLE EDT\$CHMPASTE EDT\$CHMPASTE - change mode PASTE c
ommand

.IDENT \V04-000\

.EXTRN EDT\$\$INS_STR, EDT\$\$MSG_BELL
.EXTRN EDT\$\$CHK_CC, EDT\$\$COMB_LN
.EXTRN EDT\$\$CS_DWN, EDT\$\$CS_LEFT
.EXTRN EDT\$\$CS_RIGHT, EDT\$\$SPLT_LN
.EXTRN EDT\$\$END_INS, EDT\$\$RD_CURLN
.EXTRN EDT\$\$RD_NXTLN, EDT\$\$START_INS
.EXTRN EDT\$\$STOP_BUF, EDT\$\$G_PST_CNT
.EXTRN EDT\$\$A_ALT_BUF, EDT\$\$A_CUR_BUF
.EXTRN EDT\$\$L_LNOO, EDT\$\$A_WK_LN
.EXTRN EDT\$\$G_CC_DONE, EDT\$\$L_WK_INSCNT
.EXTRN EDT\$_ATTPASCUR

.PSECT _EDT\$CODE, NOWRT, SHR, PIC, 2

.ENTRY EDT\$\$PST_CMD, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0667
R10

MOVAB EDT\$\$RD_CURLN, R10
MOVAB EDT\$\$A_WK_LN, R9
MOVAB EDT\$\$SPLT_LN, R8
MOVAB EDT\$\$A_ALT_BUF, R7
MOVAB FIRST_WORD, R6
MOVAB EDT\$\$G_PST_CNT, R5
MOVAB EDT\$\$A_CUR_BUF, R4
MOVL EDT\$\$A_ALT_BUF, R0
BEQL 1\$
CMLP R0, EDT\$\$A_CUR_BUF
BNEQ 2\$
PUSHL #EDT\$_ATTPASCUR
CALLS #1, EDT\$\$MSG_BELL
BRW 20\$
MOVL EDT\$\$A_CUR_BUF, SAVE_TBCB
CLRL EDT\$\$G_PST_CNT
ADDL3 #24, EDT\$\$A_ALT_BUF, R0
MOVZWL HIGH_1, R1
CMPW 4(R0), R1
BGTRU 3\$
BNEQ 5\$
MOVL LOW_1, R1
CMLP R1, -(R0)
BGEQU 4\$
MNEGL #1, R0
BRB 6\$
BNEQ 5\$

07FC 00000

```
5A 00000000G 00 9E 00002  
59 00000000G 00 9E 00009  
58 00000000G 00 9E 00010  
57 00000000G 00 9E 00017  
56 00000000G 00 9E 0001E  
55 00000000G 00 9E 00025  
54 00000000G 00 9E 0002C  
50 67 D0 00033  
12 13 00036  
64 50 D1 00038  
10 12 0003B  
00000000G 8F DD 0003D  
00 01 FB 00043  
00FC 31 0004A 1$:  
53 64 D0 0004D 2$:  
65 D4 00050  
67 18 C1 00052  
51 00000000G 00 3C 00056  
51 04 A0 B1 0005D  
0E 1A 00061  
17 12 00063  
51 00000000G 00 D0 00065  
60 51 D1 0006C  
05 1E 0006F  
50 01 CE 00071 3$:  
09 11 00074  
04 12 00076 4$:
```

00000000G

50

0736
0742
0745
0746
0752
0753
0758

	50	D4	00078	CLRL	R0		
	03	11	0007A	BRB	6\$		
	50	01	D0 0007C	5\$: MOVL	#1, R0		
	01	50	D1 0007F	6\$: CMPL	R0, #1		
		03	12 00082	BNEQ	7\$		
		00BE	31 00084	BRW	19\$		
00000000G	00	00	FB 00087	7\$: CALLS	#0, EDT\$\$START_INS		0760
68		00	FB 0008E	CALLS	#0, EDT\$\$SPLT_LN		0764
00000000G	00	00	FB 00091	CALLS	#0, EDT\$\$CS_LEFT		0765
		52	D4 00098	CLRL	BLANK_LINES		0767
	64	67	D0 0009A	MOVL	EDT\$\$A_ALT_BUF, EDT\$\$A_CUR_BUF		0771
00000000G	00	00	FB 0009D	CALLS	#0, EDT\$\$STOP_BUF		0772
	50	69	D0 000A4	8\$: MOVL	EDT\$\$A_WK_LN, R0		0781
		60	95 000A7	TSTB	(R0)		
		27	13 000A9	BEQL	11\$		
	64	53	D0 000AB	MOVL	SAVE TBCB, EDT\$\$A_CUR_BUF		0784
	6A	00	FB 000AE	CALLS	#0, EDT\$\$RD_CURLN		0785
		52	D5 000B1	9\$: TSTL	BLANK_LINES		0787
		07	15 000B3	BLEQ	10\$		
	68	00	FB 000B5	CALLS	#0, EDT\$\$SPLT_LN		0789
		52	D7 000B8	DECL	BLANK_LINES		0790
		F5	11 000BA	BRB	9\$		0787
	64	67	D0 000BC	10\$: MOVL	EDT\$\$A_ALT_BUF, EDT\$\$A_CUR_BUF		0793
	6A	00	FB 000BF	CALLS	#0, EDT\$\$RD_CURLN		0794
	50	69	D0 000C2	MOVL	EDT\$\$A_WK_LN, R0		0795
	7E	60	9A 000C5	MOVZBL	(R0), =(SP)		
		07	A0 9F 000C8	PUSHAB	7(R0)		
00000000G	00	02	FB 000CB	CALLS	#2, EDT\$\$INS_STR		
	50	69	D0 000D2	11\$: MOVL	EDT\$\$A_WK_LN, R0		0798
	50	60	9A 000D5	MOVZBL	(R0), R0		
	50	65	C0 000D8	ADDL2	EDT\$\$G_PST_CNT, R0		
	65	01	A0 9E 000DB	MOVAB	1(R0), EDT\$\$G_PST_CNT		
00000000G	00	00	FB 000DF	CALLS	#0, EDT\$\$RD_NXTLN		0800
	17	50	E9 000E6	BLBC	R0, 13\$		
00000000G	00	00	FB 000E9	CALLS	#0, EDT\$\$CHK_CC		0806
	09	50	E9 000F0	BLBC	R0, 12\$		
00000000G	00	01	D0 000F3	MOVL	#1, EDT\$\$G_CC_DONE		0809
		04	11 000FA	BRB	13\$		0808
		52	D6 000FC	12\$: INCL	BLANK_LINES		0813
		A4	11 000FE	BRB	8\$		0774
	64	53	D0 00100	13\$: MOVL	SAVE TBCB, EDT\$\$A_CUR_BUF		0821
	6A	00	FB 00103	CALLS	#0, EDT\$\$RD_CURLN		0822
	01	52	D1 00106	14\$: CMPL	BLANK_LINES, #1		0824
		07	15 00109	BLEQ	15\$		
	68	00	FB 0010B	CALLS	#0, EDT\$\$SPLT_LN		0826
		52	D7 0010E	DECL	BLANK_LINES		0827
		F4	11 00110	BRB	14\$		0824
		52	D5 00112	15\$: TSTL	BLANK_LINES		0830
		21	12 00114	BNEQ	17\$		
00000000G	00	00	FB 00116	CALLS	#0, EDT\$\$CS_DWN		0833
		7E	D4 0011D	CLRL	-(SP)		0834
00000000G	00	01	FB 0011F	CALLS	#1, EDT\$\$COMB_LN		
	50	66	D0 00126	MOVL	FIRST_WORD, SAVE		0835
		66	D7 00129	DECL	FIRST_WORD		
		66	D1 0012B	CMPL	FIRST_WORD, SAVE		
	50	03	1B 0012E	BLEQU	16\$		
		04	A6 B7 00130	DECW	NEXT_WORD		

EDT\$CHMPASTE
V04-000

EDT\$CHMPASTE - change mode PASTE command
EDT\$\$PST_CMD - change mode PASTE command

D 8
16-Sep-1984 00:08:07
14-Sep-1984 12:22:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMPASTE.BLI;1
Page 9 (3)

	65	D7	00133	16\$:	DECL	EDT\$\$G_PST_CNT	:	0836	
	07	11	00135		BRB	18\$	:	0830	
00000000G	00	00	FB	00137	17\$:	CALLS	#0, EDT\$\$CS_RIGHT	:	0839
00000000G	00	00	FB	0013E	18\$:	CALLS	#0, EDT\$\$END_INS	:	0841
	01	D0	00145	19\$:	MOVL	#1, R0	:	0842	
		04	00148		RET		:		
	50	D4	00149	20\$:	CLRL	R0	:	0843	
		04	0014B		RET		:		

; Routine Size: 332 bytes, Routine Base: _EDT\$CODE + 0000

: 274 0844 1
: 275 0845 1 !<BLF/PAGE>

EDT
V04

SR
F
M
C

EDT\$CHMPASTE
V04-000

EDT\$CHMPASTE - change mode PASTE command
EDT\$PST_CMD - change mode PASTE command

E 8
16-Sep-1984 00:08:07
14-Sep-1984 12:22:42

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[EDT.SRC]CHMPASTE.BLI;1 (4)

Page 10

: 277 0846 1 END
: 278 0847 1
: 279 0848 0 ELUDOM

! of module EDT\$CHMPASTE

PSECT SUMMARY

Name	Bytes	Attributes
_EDT\$CODE	332	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	41	10	40	00:00.2
_\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:CHMPASTE/OBJ=OBJ\$:CHMPASTE MSRC\$:CHMPASTE.BLI/UPDATE=(ENH\$:CHMPASTE)

: Size: 332 code + 0 data bytes
: Run Time: 00:18.6
: Elapsed Time: 00:40.4
: Lines/CPU Min: 2729
: Lexemes/CPU-Min: 10136
: Memory Used: 134 pages
: Compilation Complete

0132 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CHMMRKCHG LIS	CHMPARSEN LIS	CHMSLPPOS LIS	CHMSPLIN LIS	DATA LIS
CHMPAREN LIS	CHMSAVPOS LIS	CHMSCHSTR LIS	CHMUNDEL LIS	COMMAND LIS
CHMONSTR LIS	CHMREPOS LIS	CHMSEDEL LIS	CHMTADJ LIS	CLRKEY LIS
CHMMESS LIS	CHMPASTE LIS	CHMSAVTXT LIS	CHMNEWLEN LIS	CHMPARSE LIS
CHMSAVLIN LIS	CHMSUBS LIS			